/\*TABLES\*/

CREATE TABLE `user\_list` (

`id` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`login` VARCHAR(16) NULL,

`first\_name` VARCHAR(15) NULL,

`last\_name` VARCHAR(15) NULL,

`email` VARCHAR(319) NULL,

`phone` CHAR(13) NULL,

`password` VARCHAR(64) NULL,

`enabled` TINYINT(1) NULL,

`data\_created` TIMESTAMP NULL,

`data\_updated` TIMESTAMP NULL,

`city` VARCHAR(20) NULL,

`birth\_date` DATE NULL,

`role` ENUM(‘user’,’manager’,’admin’) NULL,

PRIMARY KEY (`id`)

)ENGINE = InnoDB;

CREATE TABLE `status` (

`id` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`code` VARCHAR(255) NOT NULL,

`name` VARCHAR(255) NULL,

`closed` TINYINT(1) NULL,

PRIMARY KEY (`id`));

CREATE TABLE `next\_status` (

`status\_id` INT UNSIGNED NOT NULL,

`nstatus\_id` INT UNSIGNED NOT NULL,

FOREIGN KEY(status\_id) REFERENCES status(id),

FOREIGN KEY(nstatus\_id) REFERENCES status(id)

);

CREATE TABLE `pet\_list` (

`id` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`owner\_id` INT UNSIGNED NOT NULL,

`name` VARCHAR(32) NULL,

`species` VARCHAR(20) NULL,

`info` VARCHAR(200) NULL,

PRIMARY KEY (`id`),

FOREIGN KEY(owner\_id) REFERENCES user\_list(id)

);

CREATE TABLE `request\_list` (

`id` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`price` DECIMAL(5,2) NULL,

`data\_created` TIMESTAMP NULL,

`data\_updated` TIMESTAMP NULL,

`description` VARCHAR(200) NULL,

`pet\_id` INT UNSIGNED NOT NULL,

`owner\_id` INT UNSIGNED NOT NULL,

`petsittler\_id` INT UNSIGNED NULL,

`status\_id` INT UNSIGNED NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY(pet\_id) REFERENCES pet\_list(id),

FOREIGN KEY(owner\_id) REFERENCES user\_list(id),

FOREIGN KEY(petsittler\_id) REFERENCES user\_list(id),

FOREIGN KEY(status\_id) REFERENCES status(id)

);

CREATE TABLE `request\_history` (

`id` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`price` DECIMAL(5,2) NULL,

`data\_updated` TIMESTAMP NULL,

`description` VARCHAR(200) NULL,

`review` VARCHAR(200) NULL,

`rating` TINYINT(1) NULL,

`pet\_id` INT UNSIGNED NOT NULL,

`owner\_id` INT UNSIGNED NOT NULL,

`petsittler\_id` INT UNSIGNED NULL,

`status\_id` INT UNSIGNED NOT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY(pet\_id) REFERENCES pet\_list(id),

FOREIGN KEY(owner\_id) REFERENCES user\_list(id),

FOREIGN KEY(petsittler\_id) REFERENCES user\_list(id),

FOREIGN KEY(status\_id) REFERENCES status(id)

);

/\*PROCEDURES\*/

/\*INSERT\*/

DELIMITER $$

CREATE PROCEDURE insert\_user\_list(IN login VARCHAR(16), IN first\_name VARCHAR(15), IN last\_name VARCHAR(15), IN email VARCHAR(319), IN phone CHAR(13), IN password VARCHAR(64), IN city VARCHAR(20), IN birth\_date DATE)

BEGIN

SET @pas=TO\_BASE64(AES\_ENCRYPT(password,'stas'));

INSERT INTO user\_list (login, first\_name, last\_name, email, phone, password,enabled,data\_created, city, birth\_date, role) VALUES(login, first\_name, last\_name, email, phone, @pas,1,CURRENT\_TIMESTAMP, city, birth\_date, 1);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE insert\_pet\_list(IN owner\_id INT, IN name VARCHAR(32), IN species VARCHAR(20), IN info VARCHAR(200))

BEGIN

INSERT INTO pet\_list (owner\_id, name, species, info) VALUES(owner\_id, name, species, info);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE insert\_request\_list(IN price DECIMAL(5,2), IN description VARCHAR(200), IN pet\_id INT, IN owner\_id INT)

BEGIN

INSERT INTO request\_list (price, description, data\_created, pet\_id , owner\_id, status\_id) VALUES(price, description, CURRENT\_TIMESTAMP, pet\_id , owner\_id, 1);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE insert\_request\_history (IN price DECIMAL(5,2), IN description VARCHAR(200), IN pet\_id INT, IN owner\_id INT, IN petsitter\_id INT, IN status\_id INT, IN data\_updated TIMESTAMP)

BEGIN

INSERT INTO request\_history (price, description, pet\_id , owner\_id, petsitter\_id, status\_id, data\_updated) VALUES(price, description, pet\_id , owner\_id, petsitter\_id, status\_id, data\_updated);

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE insert\_request\_history\_from\_list (IN newreview VARCHAR(200), IN newrating TINYINT(1), IN req\_id INT)

BEGIN

INSERT INTO request\_history (price, description, review, rating, pet\_id , owner\_id, petsittler\_id, status\_id, data\_updated) VALUES((SELECT price FROM request\_list WHERE id = req\_id), (SELECT description FROM request\_list WHERE id = req\_id), newreview, newrating, (SELECT pet\_id FROM request\_list WHERE id = req\_id), (SELECT owner\_id FROM request\_list WHERE id = req\_id), (SELECT petsittler\_id FROM request\_list WHERE id = req\_id), (SELECT status\_id FROM request\_list WHERE id = req\_id), (SELECT data\_updated FROM request\_list WHERE id = req\_id));

END$$

DELIMITER ;

/\*UPDATE\*/

DELIMITER $$

CREATE PROCEDURE update\_user\_list(IN newid INT, IN newlogin VARCHAR(16), IN newfirst\_name VARCHAR(15), IN newlast\_name VARCHAR(15), IN newemail VARCHAR(319), IN newphone CHAR(13), IN newpassword VARCHAR(64), IN newcity VARCHAR(20), IN newbirth\_date DATE)

BEGIN

UPDATE user\_list SET login = newlogin, first\_name = newfirst\_name, last\_name = newlast\_name, email = newemail, phone = newphone, password = TO\_BASE64(AES\_ENCRYPT(newpassword,'stas')), city = newcity, birth\_date = newbirth\_date WHERE id = newid;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE update\_user\_list\_name(IN newfirst\_name VARCHAR(15), IN newlast\_name VARCHAR(15), IN user\_id INT)

BEGIN

UPDATE user\_list SET first\_name = newfirst\_name, last\_name = newlast\_name WHERE id = user\_id;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE update\_request\_history\_rating(IN newreview VARCHAR(200), IN newrating TINYINT(1), IN req\_id INT)

BEGIN

UPDATE request\_history SET rating = newrating, review = newreview WHERE id = req\_id;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE update\_user\_list\_city(IN newcity VARCHAR(20), IN user\_id INT)

BEGIN

UPDATE user\_list SET city = newcity WHERE id = user\_id;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE update\_user\_list\_role(IN newrole TINYINT, IN user\_id INT)

BEGIN

UPDATE user\_list SET role = newrole WHERE id = user\_id;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE update\_user\_list\_enabled(IN newen TINYINT, IN user\_id INT)

BEGIN

UPDATE user\_list SET enabled = newen WHERE id = user\_id;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE update\_request\_list(IN newid INT, IN newprice DECIMAL(5,2), IN newdescription VARCHAR(200), IN newstatus\_id INT)

BEGIN

UPDATE request\_list SET price = newprice, description=newdescription, status\_id = newstatus\_id WHERE id = newid;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE update\_request\_list\_petsitter(IN newpetsittler\_id INT, IN request\_id INT)

BEGIN

UPDATE request\_list SET petsittler\_id = newpetsittler\_id WHERE id = request\_id;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE update\_request\_list\_status(IN newstatus\_id INT, IN request\_id INT)

BEGIN

UPDATE request\_list SET status\_id = newstatus\_id WHERE id = request\_id;

END$$

DELIMITER ;

/\*DELETE\*/

DELIMITER $$

CREATE PROCEDURE delete\_request\_list(IN newid INT)

BEGIN

DELETE FROM request\_list WHERE id = newid;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE delete\_request\_history(IN newid INT)

BEGIN

DELETE FROM request\_history WHERE id = newid;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE delete\_pet\_list(IN newid INT)

BEGIN

DELETE FROM request\_list WHERE pet\_id = newid;

DELETE FROM request\_history WHERE pet\_id = newid;

DELETE FROM pet\_list WHERE id = newid;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE delete\_user\_list(IN newid INT)

BEGIN

DELETE FROM request\_list WHERE owner\_id = newid;

DELETE FROM request\_list WHERE petsittler\_id = newid;

DELETE FROM request\_history WHERE petsittler\_id = newid;

DELETE FROM request\_history WHERE owner\_id = newid;

DELETE FROM pet\_list WHERE owner\_id = newid;

DELETE FROM user\_list WHERE id = newid;

END$$

DELIMITER ;

/\*FUNCTIONS\*/

DELIMITER $$

create function avg\_cost() RETURNS FLOAT

DETERMINISTIC

READS SQL DATA

BEGIN

SET @res= (SELECT AVG(price) FROM request\_list);

RETURN @res;

END $$

DELIMITER ;

/\*TRIGGERS\*/

DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `request\_list\_AFTER\_INSERT` AFTER INSERT ON `request\_list` FOR EACH ROW BEGIN

CALL insert\_request\_history(NEW.price, NEW.description, NEW.pet\_id , NEW.owner\_id, NULL, NEW.status\_id, NEW.data\_created);

END$$

DELIMITER ;

DELIMITER $$

CREATE DEFINER=CURRENT\_USER TRIGGER `request\_list\_AFTER\_UPDATE` AFTER UPDATE ON `request\_list` FOR EACH ROW BEGIN

IF NEW.status\_id != OLD.status\_id AND NEW.status\_id != 5 THEN CALL insert\_request\_history(NEW.price, NEW.description, NEW.pet\_id , NEW.owner\_id, NEW.petsittler\_id, NEW.status\_id, NEW.data\_updated);

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE DEFINER=CURRENT\_USER TRIGGER ` request\_list\_BEFORE\_UPDATE` BEFORE UPDATE ON `request\_list` FOR EACH ROW BEGIN

SET NEW.data\_updated = CURRENT\_TIMESTAMP;

END$$

DELIMITER ;

DELIMITER $$

CREATE DEFINER=CURRENT\_USER TRIGGER ` user\_list\_BEFORE\_UPDATE` BEFORE UPDATE ON `user\_list` FOR EACH ROW BEGIN

SET NEW.data\_updated = CURRENT\_TIMESTAMP;

END$$

DELIMITER ;